

## **REMARKS**

Reconsideration of this application as amended is respectfully requested. In the Office Action claims 1, 3-11, 13-18, 20-22, 24-33, 35-43 and 45-50 are pending. Claims 1, 3-11, 13-18, 20-22, 24-33, 35-43 and 45-50 are rejected. By way of the present response Applicant has: 1) amended claims 1, 8, 15, 20, 27, 31, 33, 40, and 47; 2) added no new claims; and 3) canceled no claims. As such, claims 1, 3-11, 13-18, 20-22, 24-33, 35-43 and 45-50 remain pending. Applicants respectfully request reconsideration of the present application and the allowance of all claims now presented. Applicant submits no new matter has been added.

### **I. CLAIM OBJECTIONS**

Claims 27-30, 33 and 35-39 have been objected to because of informalities specified in the Office Action. Office Action, May 4, 2007, p. 8.

The Office Action has objected to claim 27 as reciting poor sentence structure. Office Action, p. 8. The sentence structure in question is "logic to prune walking of nodes." While Applicant disagrees with this rejection based on common usage of the English language, Applicant has amended the claim and requests withdrawal of the claim rejection as well as the rejections of dependent claims 28-30. Additionally, Applicant submits that the amendment is non-narrowing because the amendment is solely for the purpose of reformulating sentence structure.

The Office Action has also objected to claim 33 as reciting poor sentence structure. Id. Applicant has amended claim 33 and requests withdrawal of the rejection.

## **II. REJECTIONS UNDER 35 U.S.C. § 101**

The Office Action has rejected claims 31, 32, 33, 35-39, 40-43, 45, 46, and 47-50 under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter. Office Action, pp. 8-9.

Specifically, claim 31 stands rejected because "Claim 31 appears to be no more than a program per se." Id. at p. 9. Applicant has amended claim 31 to include the limitation, "hardware" and respectfully requests withdrawal of the claim rejection as well as the rejection of dependent claim 32.

As for claim 33, 35-39, 40-43, 45, 46, and 47-50, Applicant has amended the claims to include the limitation "hardware" and respectfully requests withdrawal of the claim rejections.

## **III. Office Action's Response to Arguments**

The Office Action has included a "Response to Arguments" section. See Office Action, pp. 2-7. With this present response, Applicant has not addressed each and every argument put forth in the Office Action. However, Applicant submits that this is not to be construed in any way as acquiescence to the Office Action's arguments.

Regarding the Office Action's argument that Zwilling performs "garbage collection," Applicant submits that Zwilling does not perform garbage collection as the term is used in the computer science art. Garbage collection is a form of automatic removal of garbage from a memory space. Additionally, garbage, in the context of the computer sciences art, refers to any object or data within a program's memory space that is unreachable from the program's root set and cannot ever be accessed by a running program. As such, it resides in memory, but is garbage because it cannot be used in any program within the memory space. The Office Action argues, "Zwilling also does garbage collection ... because used parts of a file are moved to unused allocated parts of the file and any unreferenced parts of a file are deallocated. This enables a smaller size for a file and can be seen as removing the garbage (unreferenced and allocated units) from a file (Zwilling, col. 5, lines 29-52)." The problem with the Office Action's argument is that it views removing empty space as garbage collecting because Zwilling shrinks files by removing empty space. If this were the case, then the empty space would be the garbage that is being removed. However, as discussed above, garbage refers to *objects* or *data* not empty space. In Zwilling the empty space is being filled with allocation units that contain data. This results in a compaction of the file. Removing empty space from a file cannot ever be the same thing as collecting garbage because the empty space is not "garbage" as is known in the art. Garbage under the understanding of a person of ordinary skill in the art is "filled space" such as objects or data. It is desirable to remove these objects or data because they are garbage and cannot ever be accessed by a running program. Thus, a garbage

collector, as is known in the art, collects "garbage," otherwise known as objects or data. A simple analogy would be garbage that is left out at the end of a person's driveway. If garbage is placed out on one's driveway at the correct time, then it will likely be picked up by the garbage man. If there is empty space out where the garbage would be placed, then there is no need to collect it because there is no garbage to collect. Zwilling removes empty space which cannot ever be objects or data. Therefore, Applicant submits that Zwilling does not perform garbage collection as required by the claims. Applicant respectfully requests withdrawal of the relevant claim rejections.

Additionally, the Office Action asserts that Zwilling teaches, "[a] method of garbage collection in a storage device ..." Office Action, p. 10. In support of this the Office Action argues, "Zwilling must have a determination of a blocks being used/active/reference and unused/inactive/unreferenced in order to copy the allocation units and guarantee file integrity." Office Action, pp. 4-5. The Office Action further states, "[t]his is shown in Zwilling, col. 12, line 37 'not in use' and Zwilling, col. 5, lines 37 'if an allocation unit ... is used.'" However, the Office Action's argument is conflating two distinctly different concepts. The Office Action is equating the concept of "used" with the concept of "active/referenced" by using the term, "used/active/referenced." Additionally, the Office Action is equating the concept of "unused" with "inactive/unreferenced" by using the term, "unused/inactive/unreferenced." However, Applicant submits that these two concepts cannot be properly equated to read on claim 1.

The concept of active/referenced blocks of data or inactive/unreferenced blocks within a backup system is well-known in the art. When snapshots of a file system are taken and stored on a backup storage device there will eventually be differences between snapshots. This is because files will be added to and deleted from the file system between snapshots resulting in differences between the snapshots taken at different times. In order to decrease memory-space utilization, some backup systems only store a block of data once, and then create pointers to the originally stored block in subsequent snapshots. Thus, older blocks of data that remain within the active file system when a new snapshot is taken become part of the new snapshot, but only as a pointer to the original data block that was physically stored earlier on the storage device. When a data block is deleted from the active file system it will not be deleted from the storage device because the storage device contains large amounts of data and it would be too costly/time consuming to search for the deleted data in the storage device. Instead, the backup system just removes the data block from any subsequent snapshots. As a result, there will no longer be to the originally stored block of data that was deleted on the active file system references in any new snapshots. As older snapshots on the system are deleted, the block eventually becomes unreferenced by any of the snapshots maintained on the backup system. In this case, the block of data is just sitting there taking up space. Once this occurs, it is desirable to delete this block because it is useless to any program running on the machine. This is what is referred to in the art as "unreferenced" data blocks.

Applicant submits that this has nothing to do with whether or not the data block is empty. These data blocks are garbage, and by definition (see above), continue to contain data. That is, the data blocks containing garbage are not scheduled for deallocation because they are empty; but rather, are scheduled for deallocation because they are garbage.

As a result, a garbage collection mechanism is needed to go through and figure out which blocks within the storage system are no longer referenced and clean them out. This is a distinctly different concept from the one asserted by the Office Action based on the Zwilling reference. The concept described in Zwilling is based on removing empty space by deleting data blocks (allocation units) in files that are no longer “*used*.” In this case, “used” refers to whether the block contains data. In Zwilling, the data in allocated blocks above the fence is copied to an unallocated block below the fence and the block is marked as unallocated. However, as discussed above, data blocks that are “unreferenced” contain data within them. Thus, whether a data block is used or unused is *not* the same thing as whether a data block is referenced/active or unreferenced/inactive as required by claim 1. A person of ordinary skill in the art would recognize this distinction. Accordingly, Zwilling fails to teach or fairly suggest “[a] method of garbage collection in a storage device ...” as required by claim 1.

For these reasons Applicant submits that Zwilling cannot be properly used as the primary reference for this case and respectfully requests withdrawal of the claim rejections containing this limitation.

#### **IV. REJECTIONS UNDER 35 U.S.C. § 103 OVER ZWILLING IN VIEW OF HITZ.**

The Office Action has rejected claims 1, 3-7, 18, 20-22, 24-33, 35-39, and 50 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,249,792 to Zwilling, et al., ("Zwilling") in view of U.S. Patent No. 5,963,962 to Hitz, et al. ("Hitz"). Office Action, pp. 9-31. Applicant does not admit that Hitz and Zwilling are prior art and reserves the right to swear behind either reference at a later date.

##### **a. Independent claims 1, 20, 27, 31, and 33**

The Office Action has rejected independent claims 1, 20, 27, 31, and 33 under §103 as being unpatentable over Zwilling in view of Hitz. Office Action, p. 10. Applicant respectfully disagrees with the Office Action's basis for rejecting the claims and submits that claims 1, 20, 27, 31, and 33 are patentable over Zwilling in view of Hitz.

##### **A. The combination does not read on claims 1, 20, 27, 31 and 33**

The Office Action has indicated that alternative #2 (Zwilling to shrink files before applying Hitz to take snapshots of the shrunk files) in the Applicant's previous remarks is the correct paradigm for considering the Zwilling and Hitz references in combination. Office Action, pp. 2-3. As a result, Applicant submits the following remarks in the light of combination alternative #2.

The Office Action admits, "Zwilling does not expressly disclose storage trees each representing a snapshot and where unreferenced data remains untouched."

Office Action, p. 12. However, the Office Action states, "Hitz discloses a write anywhere file-system layout comprising storage trees (file system snapshot trees) each representing a snapshot and where unreferenced data remains untouched." Id. The Office Action argues, "[i]t would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to take the write anywhere file-system layout from Hitz and install it into the invention of Zwilling, thereby offering the obvious advantage of *taking snapshots of garbage collected/shrunken data*, thus saving space. Shrinking the files *prior* to snapshotting them frees more space in Hitz thus creating more room in Hitz so that another later snapshot won't be prematurely forced out when space in Hitz runs out (by snapshots consuming unacceptable numbers of disk blocks). Office Action, p. 12. Thus, the Office Action's argument can be summarized to say that Zwilling in view of Hitz renders claims 1, 20, 27, 31, and 33 unpatentable under § 103 when the garbage collection is performed *before* the snapshots are taken. That is, the garbage is collected and then the snapshots are taken of the garbage collected data.

Applicant submits that taking snapshots of garbage collected data does not anticipate claims 1, 20, 27, 31, and 33 because the claims expressly require the opposite order of operations of performing garbage collection on snapshots. For example, claim 1 (as amended) recites the following:

1. A method of garbage collecting in a storage device comprising:  
locating blocks of data in a log that are both referenced by at least one other block of data residing within a set of one or more storage trees, and within a range at a tail of the log using



pruned walking, the range representing an address range within an allocated segment of the log, wherein the log is implemented in a hierarchical architecture having a plurality of storage trees, each storage tree representing a snapshot taken at a point in time of target data being processed, each storage tree having a plurality of nodes and each node representing a block of data of the snapshot associated with each storage tree;

copying the blocks of data that are referenced by one or more other blocks of data of other nodes and within the range to an unallocated segment of the log, wherein blocks of data that are not referenced by other blocks of data and within the range remain untouched, and

marking the range at the tail of the log as unallocated so that at least a portion of an address space within the range can be reclaimed.

Claims 20, 27, 31, and 33 contain similar limitations.

Applicant points out that claim 1 specifically requires both the limitation, "locating blocks of data in a log that are referenced ... ," and the limitation, "wherein the log is implemented in a hierarchical architecture having a plurality of storage trees, each storage tree representing a snapshot taken at a point in time of target data being processed, each storage tree having a plurality of nodes and each node representing a block of data of a snapshot associated with each storage tree."

Applicant submits that the garbage collection operation claimed in claim 1 is performed on the already created snapshots. Specifically, claim 1 performs "locating blocks of data in a log ... wherein the log is implemented in a hierarchical architecture having a plurality of storage trees, each storage tree representing a snapshot taken at a point in time ...." It is clear, based on the claim language of claim 1, that the garbage collection is not performed before the snapshots are taken as asserted by the Office Action. See Office Action, pp. 2-3, 12. That is, claim 1 specifically requires garbage collecting the data that is already part of a storage tree

representing a snapshot at a point in time. As such, the garbage collection cannot be performed before each snapshot is taken. Therefore, Applicant submits that under the alternative #2 paradigm, Zwilling and Hitz cannot read on claims 1, 20, 27, 31 or 33. Accordingly, Applicant respectfully requests withdrawal of the rejections of claims 1, 20, 27, 31, and 33 as well as the rejections of their associated dependent claims.

Additionally, although Applicant disagrees with the basis for the Office Action's rejection, Applicant has amended claims 1, 20, 27, 31 and 33. As a result, Applicant submits that the combination of Zwilling and Hitz fails to at least disclose the following bolded limitations:

1. A method of garbage collecting in a storage device comprising:
  - locating blocks of data in a log that are **both referenced by at least one other block of data residing within a set of one or more storage trees**, and within a range at a tail of the log using pruned walking, the range representing an address range within an allocated segment of the log, wherein the log is implemented in a hierarchical architecture having a plurality of storage trees, each storage tree representing a snapshot taken at a point in time of target data being processed, each storage tree having a plurality of nodes and each node representing a block of data of the snapshot associated with each storage tree;
  - copying the blocks of data that are referenced by one or more other blocks of data of other nodes and within the range to an unallocated segment of the log, wherein blocks of data that are not referenced by other blocks of data and within the range remain untouched, and
  - marking the range at the tail of the log as unallocated so that at least a portion of an address space within the range can be reclaimed.

Claims 1, 20, 27, 31 and 33 now expressly require data blocks to be referenced by at least one other data block residing within the storage trees. Rather, Zwilling

locates “used” blocks (see discussion above) and copies the data within those blocks to an unallocated block, and then deallocates the block to compact a file. Applicant submits that this is not what is required by a system that complies with claim 1 and respectfully requests withdrawal of the claim rejections.

**V. REJECTIONS UNDER 35 U.S.C. § 103 OVER DIFFERENT EMBODIMENTS OF ZWILLING**

The Office Action has rejected claims 8-17, 19, 40-49, and 51 under 35 U.S.C. §103(a) as being unpatentable over Zwilling. It is respectfully submitted that claims 8-17, 19, 40-49, and 51 include limitations that are not disclosed or suggested by Zwilling.

**a. Independent claims 8, 15, 40 and 47**

The Office Action has also rejected claims 8, 15, 40 and 47 under § 103 over Zwilling. However, Applicant respectfully disagrees and submits the following remarks in support of Applicant’s position. Applicant submits that Zwilling fails to disclose at least the following bolded limitations in claim 8 as amended:

8. A method comprising:  
***garbage collecting*** within a range of addresses in a storage system having a plurality of storage trees, each storage tree having a plurality of nodes and having multiple references to a same block of data, the garbage collecting including:  
pruning walking of the plurality of storage trees to determine active blocks of data within said range, where active blocks of data are those ***that are referenced by at least one other block of data still in one of the***

***plurality of storage trees***, the pruning walking including:

determining, based on accessing in one of said plurality of storage trees a parent node that has a plurality of descendent nodes, that none of the plurality of descendant nodes are associated with blocks of data within the range; and

skipping the walking of the plurality of descendent nodes based on said determining,

wherein the active blocks determined to be in the range are copied out of the range and the range is marked as unallocated so that at least a portion of the address space within the range can be reclaimed.

Claims 15, 40 and 47 have been amended to include similar limitations. Applicant submits that Zwilling fails to describe "blocks of data ... that are referenced by at least one other block of data still in one of the plurality of storage trees" as required by claims 8, 15, 40 and 47 for the same reasons as articulated above with respect to claim 1. Accordingly, Applicant respectfully requests withdrawal of the claim rejection as well as the rejections of the claims which depend, either directly or indirectly, on any of claims 8, 15, 40 and 47, and incorporate all the limitations contained therein.

**CONCLUSION**

Applicant respectfully submits that all rejections have been overcome and that all pending claims are in condition for allowance. If there are any additional charges, please charge them to our Deposit Account Number 02-2666. If a telephone conference would facilitate the prosecution of this application, Examiner is invited to contact Matthew W. Hindman at (408) 720-8300.

Respectfully submitted,  
**BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN**

Date: 8-14-07

A handwritten signature in black ink, appearing to read 'Matthew W. Hindman', written over a horizontal line.

Matthew W. Hindman  
Attorney at Law  
Reg. No. 57,396  
matthew\_hindman@bstz.com

1279 Oakmead Pkwy  
Sunnyvale, CA 94085-4040  
(408) 720-8300